Complexity in Convex Languages

Janusz Brzozowski

David R. Cheriton School of Computer Science, University of Waterloo,
Waterloo, ON, Canada N2L 3G1
brzozo@uwaterloo.ca

Abstract. A language L is prefix-convex if, whenever words u and w are in L with u a prefix of w, then every word v which has u as a prefix and is a prefix of w is also in L. Similarly, we define suffix-, factor-, and subword-convex languages, where by subword we mean subsequence. Together, these languages constitute the class of convex languages which contains interesting subclasses, such as ideals, closed languages (including factorial languages) and free languages (including prefix-, suffix-, and infix-codes, and hypercodes). There are several advantages of studying the class of convex languages and its subclasses together. These classes are all definable by binary relations, in fact, by partial orders. Closure properties of convex languages have been studied in this general framework of binary relations. The problems of deciding whether a language is convex of a particular type have been analyzed together, and have been solved by similar methods. The state complexities of regular operations in subclasses of convex languages have been examined together, with considerable economies of effort. This paper surveys the recent results on convex languages with an emphasis on complexity issues.

Keywords: automaton, bound, closed, complexity, convex, decision problem, free, ideal, language, quotient, regular, state complexity

1 Convex Languages

We begin by defining our terminology and notation. If Σ is a non-empty finite *alphabet*, then Σ^* is the free monoid generated by Σ. A *word* is any element of Σ^*, and the *empty word* is ε. A *language* over Σ is any subset of Σ^*.

If $u, v, w, x \in \Sigma^*$ and $w = uxv$, then u is a *prefix* of w, x is a factor of w, and v is a *suffix* of w. A prefix or suffix of w is also a factor of w. If $w = w_0a_1w_1 \cdots a_nw_n$, where $a_1, \ldots, a_n \in \Sigma$, and $w_0, \ldots, w_n \in \Sigma^*$, then $x = a_1 \cdots a_n$ is a *subword* of w. Every factor of w is also a subword of w.

Definition 1. *A language L is* prefix-convex *if $u, w \in L$ with u a prefix of w implies that every word v must also be in L if u is a prefix of v and v is a prefix of w; L is* prefix-free *if $w \in L$ implies that no prefix of w other than w is in L; L is* prefix-closed *if $w \in L$ implies that every prefix of w is in L; L is* converse prefix-closed *if $w \in L$ implies that every word that has w as a prefix is also in L.*

In a similar way, we define *suffix-convex*, *factor-convex*, and *subword-convex* languages, and the corresponding free, closed, and converse closed versions.

A language is *bifix-convex* (respectively, *bifix-free* or *bifix-closed*) if it is both prefix- and suffix-convex (respectively prefix- and suffix-free or prefix- and suffix-closed). The class of bifix-closed languages coincides with the class of factor-closed languages [1].

Definition 2. *A language $L \subseteq \Sigma^*$ is a* right ideal *(respectively,* left ideal*,* two-sided ideal*) if it is non-empty and satisfies $L = L\Sigma^*$ (respectively, $L = \Sigma^* L$, $L = \Sigma^* L \Sigma^*$). A two-sided ideal which satisfies the condition $L = \Sigma^* \sqcup L = \bigcup_{a_1 \cdots a_n \in L} \Sigma^* a_1 \Sigma^* \cdots \Sigma^* a_n \Sigma^*$, where $\sqcup$ is the shuffle operator, is called an* all-sided ideal*. We refer to all four types as* ideal languages *or simply* ideals*.*

Ideals and closed languages are related as follows [1]: A non-empty language is a right ideal (respectively, left, two-sided, or all-sided ideal ideal) if and only if its complement is not Σ^* and is prefix-closed (respectively, suffix-, factor-, or subword-closed).

Here is a brief history of the work on the class of convex languages and its subclasses; for more details see [1]. To avoid making many definitions of other authors' terms, we use our own terminology when discussing previous results.

Sets that are closed with respect to an arbitrary reflexive and transitive binary relation were introduced in 1952 by Higman [15]. His results were rediscovered several times in various contexts; a detailed account of this history was given by Kruskal [19] in 1972. Left and right ideals were studied by Paz and Peleg [22] in 1965 under the names "ultimate definite" and "reverse ultimate definite events". In 1969 Haines [12] examined subword-free, subword-closed and converse subword-closed languages, and also used all-sided ideals. Subword-convex languages were introduced by Thierrin [26] in 1973, who also studied subword-closed and converse subword-closed languages. Subword-free languages were studied by Shyr and Thierrin in 1974 under the name of hypercodes [25]. Suffix-closed languages were examined by Gill and Kou in 1974 [11]. Prefix-free and suffix-free languages (codes) were studied in depth in the 1985 book of Berstel and Perrin [2] and also in an updated version of the book [3]. In 1990, de Luca and Varricchio [20] observed that a language is factor-closed if and only if it is the complement of a two-sided ideal. In 1991 Jürgensen and Yu [17] studied codes that are free languages with respect to many binary relations, including the prefix, suffix, factor and subword relations; that paper also contains additional references to codes. More information about codes can be found in the 1997 article by Jürgensen and Konstantinidis [16]. In 2001 Shyr [24] studied right, left, and two-sided ideals and their generators in connection with codes.

Most of the material in this paper is based on recent work on convex languages [1, 5–9, 13, 14, 18, 27, 28]. The remainder of the paper is organized as follows: In Section 2 we define convex languages in terms of partial orders. Closure properties of convex languages are discussed in Section 3. The complexity of decision problems about convex languages is presented in Section 4. Quotient complexity (which is equivalent to state complexity) is defined in Section 5. The

quotient complexity of boolean operations in convex languages is then covered in Section 6. Section 7 summarizes the known results for the complexity of product, star, and reversal in convex languages. The special case of unary convex languages is treated in Section 8. The complexity of operations in closed and ideal language classes is discussed in Section 9, and Section 10 closes the paper.

2 Languages Defined by Partial Orders

For further details on the material in this section see [1]. The relations used in this paper to define classes of languages are all partial orders. Let $\trianglelefteq$ be an arbitrary partial order on Σ^*. If $u \trianglelefteq v$ and $u \neq v$, we write $u \triangleleft v$. Let $\trianglerighteq$ be the converse binary relation, that is, let $u \trianglerighteq v$ if and only if $v \trianglelefteq u$.

Definition 3. *A language L is* $\trianglelefteq$-convex *if $u \trianglelefteq v$ and $v \trianglelefteq w$ with $u, w \in L$ implies $v \in L$. It is* $\trianglelefteq$-free *if $v \triangleleft w$ and $w \in L$ implies $v \notin L$. It is* $\trianglelefteq$-closed *if $v \trianglelefteq w$ and $w \in L$ implies $v \in L$. It is* $\trianglerighteq$-closed *if $v \trianglerighteq w$ and $w \in L$ implies $v \in L$.*

For an arbitrary partial order $\trianglelefteq$ on Σ^* and a language $L \subseteq \Sigma^*$, define the *closure* ${}_{\trianglelefteq}L$ and *converse closure* $L_{\trianglelefteq}$ of L:

$$ {}_{\trianglelefteq}L = \{v \in \Sigma^* \mid v \trianglelefteq w \text{ for some } w \in L\}, $$

$$ L_{\trianglelefteq} = \{v \in \Sigma^* \mid w \trianglelefteq v \text{ for some } w \in L\}. $$

The following are consequences of the definitions:

Proposition 1. *Let $\trianglelefteq$ be an arbitrary partial order on Σ^*. Then*

1. *A language is $\trianglelefteq$-convex if and only if it is $\trianglerighteq$-convex.*
2. *A language is $\trianglelefteq$-free if and only if it is $\trianglerighteq$-free.*
3. *Every $\trianglelefteq$-free language is $\trianglelefteq$-convex.*
4. *Every $\trianglelefteq$-closed language and every $\trianglerighteq$-closed language is $\trianglelefteq$-convex.*
5. *A language is $\trianglelefteq$-closed if and only if its complement is $\trianglerighteq$-closed.*
6. *A language L is $\trianglelefteq$-closed ($\trianglerighteq$-closed) if and only if $L = {}_{\trianglelefteq}L$ ($L = L_{\trianglelefteq}$).*

The following special cases are of interest to us:

- Let $\leqslant$ denote the partial order "is a prefix of". If $\trianglelefteq$ is $\leqslant$, then we get prefix-convex, prefix-free, prefix-closed, and right ideal languages.
- Let $\preceq$ denote the partial order "is a suffix of". If $\trianglelefteq$ is $\preceq$, then we get suffix-convex, suffix-free, suffix-closed, and left ideal languages.
- Let $\sqsubseteq$ denote the partial order "is a factor of". If $\trianglelefteq$ is $\sqsubseteq$, then we get factor-convex, factor-free, factor-closed, and two-sided ideal languages.
- Let $\Subset$ denote the partial order "is a subword of". If $\trianglelefteq$ is $\Subset$, then we get subword-convex, subword-free, subword-closed, and all-sided ideal languages.

3 Closure Properties

This section is based on [1]. The following set operations are defined on languages: *complement* ($\overline{L} = \Sigma^* \setminus L$), *union* ($K \cup L$), *intersection* ($K \cap L$), *difference* ($K \setminus L$), and *symmetric difference* ($K \oplus L$). A general *boolean operation* with two arguments is denoted by $K \circ L$. We also define the *product*, usually called *concatenation* or *catenation*, ($KL = \{w \in \Sigma^* \mid w = uv, u \in K, v \in L\}$), *star* ($L^* = \bigcup_{i \geqslant 0} L^i$), and *positive closure* ($L^+ = \bigcup_{i \geqslant 1} L^i$). The reverse w^R of a word $w \in \Sigma^*$ is defined as follows: $\varepsilon^R = \varepsilon$, and $(wa)^R = aw^R$. The *reverse* of a language L is denoted by L^R and defined as $L^R = \{w^R \mid w \in L\}$. The *left quotient* of a language L by a word w is the language $L_w = \{x \in \Sigma^* \mid wx \in L\}$. The *right quotient* of L by w is the language $\{x \in \Sigma^* \mid xw \in L\}$.

It was shown in [1] that closure properties of convex languages can be studied in a common framework of binary relations. We now give some examples.

Example 1. In this example there are no conditions on the partial order $\trianglelefteq$. If $K, L \subseteq \Sigma^*$ are $\trianglelefteq$-convex ($\trianglelefteq$-free, or $\trianglelefteq$-closed), then so is $M = K \cap L$. It follows then that prefix-, suffix-, factor-, and subword-convex classes are closed under intersection, as are the corresponding free and closed versions.

Example 2. Here a condition on the partial order $\trianglelefteq$ is needed. A partial order $\trianglelefteq$ is *factoring* if $x \trianglelefteq y_1 y_2$ implies that $x = x_1 x_2$ for some $x_1, x_2 \in \Sigma^*$ such that $x_1 \trianglelefteq y_1$, $x_2 \trianglelefteq y_2$. If $\trianglelefteq$ is factoring and K and L are $\trianglelefteq$-closed, then so is KL. One then verifies that $\leqslant$, $\preceq$, $\sqsubseteq$ and $\Subset$ are factoring. From this it follows that the prefix-, suffix-, factor-, and subword-closed classes are closed under product.

Table 1 summarizes the closure results. In case closure holds only for some of the relations, these relations are specified in the table. In [1] it was incorrectly stated that free languages are closed under inverse homomorphism.

Table 1. Closure in classes defined by prefix, suffix, factor, and subword relations

	convex	*closed*	*converse closed*	*free*
intersection	yes	yes	yes	yes
union	no	yes	yes	no
complement	no	no	no	no
product	no	yes	yes	yes
Kleene star	no	yes	no	no
positive closure	no	yes	yes	no
left quotient	prefix	prefix	prefix	prefix
	subword	subword	subword	subword
right quotient	suffix	suffix	suffix	suffix
	subword	subword	subword	subword
homomorphism	no	no	no	no
inverse homomorphism	yes	yes	yes	no

4 Complexity of Decision Problems

The results of this section are from [8]. *Regular languages* over an alphabet Σ are languages that can be obtained from the *basic languages* $\emptyset$, $\{\varepsilon\}$, and $\{a\}$, $a \in \Sigma$, using a finite number of operations of union, product and star. Such languages are usually denoted by regular expressions. If E is a regular expression, then $\mathcal{L}(E)$ is the language denoted by that expression. For example, $E = (\varepsilon \cup a)^*b$ denotes $L = (\{\varepsilon\} \cup \{a\})^*\{b\}$. We use the regular expression notation for both expressions and languages. In contrast to other authors, we prefer to use the same operator symbol in expressions as in languages. Thus $\cup$ denotes union, juxtaposition denotes product, and * denotes the star.

A *deterministic finite automaton* (DFA) is a quintuple $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite, non-empty set of *states*, Σ is a finite, non-empty *alphabet*, $\delta : Q \times \Sigma \to Q$ is the *transition function*, $q_0 \in Q$ is the *initial state*, and $F \subseteq Q$ is the set of *final states*.

As usual, a *nondeterministic finite automaton* (NFA) is a quintuple $\mathcal{N} = (Q, \Sigma, \eta, S, F)$, where Q, Σ, and F are as in a DFA, $\eta : Q \times \Sigma \to 2^Q$ is the *transition function*, and $S \subseteq Q$ is the *set of initial states*. If η also allows ε as input, then we call $\mathcal{N}$ an ε-NFA.

The following questions were studied in [8]: If a regular language L is specified by a DFA, how difficult is it to decide whether L is prefix-, suffix-, factor-, or subword-convex, or -free, or -closed?

The most difficult case is that of factor-convexity. To solve this problem we test whether L is *not* factor-convex, in which case there exist $u, v, w \in \Sigma^*$, such that $u \sqsubseteq v \sqsubseteq w$, with $u, w \in L$ and $v \notin L$. Then there exist u', u'', v', v'' such that $v = u'uu''$ and $w = v'vv'' = v'u'uu''v''$.

Suppose $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$ is a DFA accepting L. We define an ε-NFA $\mathcal{N} = (Q', \Sigma, \delta', \{q_0'\}, F')$, where $Q' = Q \times Q \times Q \times \{1, 2, 3, 4, 5\}$, $q_0' = [q_0, q_0, q_0, 1]$, $F' = F \times (Q \setminus F) \times F \times \{5\}$, and δ' is defined below. States of $\mathcal{N}$ are quadruples, where components 1, 2, and 3 keep track of the state of $\mathcal{N}$ as it is processing w, v, and u, (respectively). The last component represents the *mode* of operation of $\mathcal{N}$.

The ε-NFA starts with the first three components in the initial state q_0, and the fourth in mode 1. Recall that the input string we are examining has the form $w = v'u'uu''v''$. The automaton $\mathcal{N}$ operates in mode 1 for a while, reading the input only in component 1 using $\delta'([p, q_0, q_0, 1], a) = \{[\delta(p, a), q_0, q_0, 1]\}$, for all $p \in Q, a \in \Sigma$. Then it guesses nondeterministically that it has finished reading v', and switches to mode 2 using $\delta'([p, q_0, q_0, 1], \varepsilon) = \{[p, q_0, q_0, 2]\}$, for all $p \in Q$. In mode 2, component 1 continues to read the input $u'uu''v''$ from the state reached by v', while component 2 reads that input from state q_0 using $\delta'([p, q, q_0, 2], a) = \{[\delta(p, a), \delta(q, a), q_0, 2]\}$, for all $p, q \in Q, a \in \Sigma$. At some point $\mathcal{N}$ guesses that u' has been read and switches to mode 3 using the rule $\delta'([p, q, q_0, 2], \varepsilon) = \{[p, q, q_0, 3]\}$, for all $p, q \in Q$. Now all three components read the input $uu''v'$, the first starting from the state reached by $v'u'$, the second from the state reached by u' and the third from q_0, using $\delta'([p, q, r, 3], a) = \{[\delta(p, a), \delta(q, a), \delta(r, a), 3]\}$, for all $p, q, r \in Q, a \in \Sigma$. A guess is then made us-

ing $\delta'([p,q,r,3],\varepsilon) = \{[p,q,r,4]\}$, for all $p,q,r \in Q$, that u has been read. In mode 4, component 3 stops reading since its job was to read u, and that has been done. The first two components continue to read $u''v''$, using $\delta'([p,q,r,4],a) = \{[\delta(p,a),\delta(q,a),r,4]\}$, for all $p,q,r \in Q, a \in \Sigma$. Then $\mathcal{N}$ guesses that u'' has been read, using $\delta'([p,q,r,4],\varepsilon) = \{[p,q,r,5]\}$, for all $p,q,r \in Q$. Now component 2 has finished reading v, and only component 1 continues to read the input v'' in mode 5, to finish processing w; this uses the rule $\delta'([p,q,r,5],a) = \{[\delta(p,a),q,r,5]\}$, for all $p,q,r \in Q, a \in \Sigma$. The input w is accepted by $\mathcal{N}$ if and only if $u,w \in L$ and $v \notin L$. Hence L is factor-convex if and only if $\mathcal{N}$ accepts the empty language.

One verifies that $\mathcal{N}$ has $3n^3+n^2+n$ reachable states and $(3|\Sigma|+2)n^3 + (|\Sigma|+1)(n^2+n)$ transitions, where $|\Sigma|$ is the cardinality of Σ. Since we can test for the emptiness of the language accepted by $\mathcal{N}$ using depth-first search in time linear in the size of $\mathcal{N}$, we can decide if a given regular language L accepted by a DFA with n states is factor-convex in $O(n^3)$ time, assuming $|\Sigma|$ is a constant.

To test for prefix-convexity of L, we construct an ε-NFA $\mathcal{N}$ that checks whether any word of the form $w = uu'u''$ is accepted, where $u,w \in L$ and $v = uu' \notin L$. Then L is prefix-convex if and only if $\mathcal{N}$ accepts the empty language. This construction is very similar to that for factor convexity, and the decision about prefix-convexity can also be done in $O(n^3)$ time.

For suffix convexity of L, we construct $\mathcal{N}$ to test whether $w = u''u'u$ is accepted, where $u,w \in L$ and $v = u'u \notin L$. This can be done in $O(n^3)$ time.

For subword convexity we use an NFA $\mathcal{N} = (Q', \Sigma, \delta', q'_0, F')$, where $Q' = Q \times Q \times Q$, $q'_0 = [q_0,q_0,q_0]$, $F' = F \times (Q \setminus F) \times F$, and

$$\delta'([p,q,r],a) = \{[\delta(p,a),q,r],\ [\delta(p,a),\delta(q,a),r],\ [\delta(p,a),\delta(q,a),\delta(r,a)]\},$$

for all $p,q,r \in Q$ and $a \in \Sigma$. The test can again be done in $O(n^3)$ time.

The properties of closure, freeness and converse closure can be decided in $O(n^2)$ time using similar methods.

A related problem studied in [8] is to find the length of a shortest word (*witness*) demonstrating that a language is not convex, not closed or not free. The results summarized in Table 2 are best possible, except in the case of subword convexity, where it is not known whether the bound can be reached.

Table 2. Sizes of shortest witnesses denying convexity, closure and freeness

	convexity	closure	freeness
factor	$\Theta(n^3)$	$\Theta(n^2)$	$\Theta(n^2)$
prefix	$2n-1$	n	$2n-1$
suffix	$\Theta(n^3)$	$\Theta(n^2)$	$\Theta(n^2)$
subword	$3n-2$	n	$2n-1$

The complexities of the convexity, closure, and freeness decision problems under the assumption that the language is specified by other means are also considered in [8]. For languages specified by NFA's or regular expressions, the convexity and closure problems are PSPACE-complete, but for an NFA with n states and t transitions, freeness can be decided in $O(n^2+t^2)$ time. For additional references to these problems, see the bibliography in [8].

5 Quotient Complexity

For a detailed discussion of general issues concerning state and quotient complexity see [5, 27] and the reference lists in those papers.

The *quotient complexity* of L is the number of distinct left quotients of L, and is denoted by $\kappa(L)$. From now on we refer to left quotients simply as quotients.

We now describe the computation of quotients of a regular language. First, the *ε-function* L^ε of a regular language L is equal to ε if $\varepsilon \in L$, and to $\emptyset$ otherwise. The quotient by a letter a in Σ is computed by structural induction: $b_a = \emptyset$ if $b \in \{\emptyset, \varepsilon\}$ or $b \in \Sigma$ and $b \neq a$, and $b_a = \varepsilon$ if $b = a$; $(\overline{L})_a = \overline{L_a}$; $(K \cup L)_a = K_a \cup L_a$; $(KL)_a = K_a L \cup K^\varepsilon L_a$; $(K^*)_a = K_a K^*$. The quotient by a word $w \in \Sigma^*$ is computed by induction on the length of w: $L_\varepsilon = L$; $L_w = L_a$ if $w = a \in \Sigma$; and $L_{wa} = (L_w)_a$. Quotients computed this way are indeed the left quotients of L [4, 5]. A quotient L_w is *accepting* if $\varepsilon \in L_w$; otherwise it is *rejecting*.

The *quotient automaton* of a regular language L is $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{L_w \mid w \in \Sigma^*\}$, $\delta(L_w, a) = L_{wa}$, $q_0 = L_\varepsilon = L$, $F = \{L_w \mid L_w^\varepsilon = \varepsilon\}$, and $L_w^\varepsilon = (L_w)^\varepsilon$. So the number of states in the quotient automaton of L is the quotient complexity of L. The *state complexity of a regular language* L is the number of states in the minimal DFA recognizing L. Evidently, the quotient complexity of L is equal to state complexity of L.

The *quotient complexity of a regular operation in* $\mathbb{Q}$ is defined as the worst case size of the quotient automaton for the language resulting from the operation, taken as a function of the quotient complexities of the operands in $\mathbb{Q}$.

Although quotient and state complexities are equal, there are advantages in using quotients. The following formulas [4, 5] for quotients of regular languages can be used to establish upper bounds on quotient complexity of operations:

Proposition 2. *If K and L are regular languages, $u, v \in \Sigma^+$, $w \in \Sigma^*$, then*

$$(\overline{L})_w = \overline{L_w}, \quad (K \circ L)_w = K_w \circ L_w, \tag{1}$$

$$(KL)_w = K_w L \cup K^\varepsilon L_w \cup \Big(\bigcup_{w=uv} K_u^\varepsilon L_v \Big), \tag{2}$$

$$(L^*)_\varepsilon = \varepsilon \cup LL^*, \quad (L^*)_w = \Big(L_w \cup \bigcup_{w=uv} (L^*)_u^\varepsilon L_v\Big) L^* \text{ for } w \in \Sigma^+. \tag{3}$$

In the sequel we consider the quotient complexity of operations in various classes of convex languages.

6 Quotient Complexity of Boolean Operations

Table 3 shows the complexities of union and intersection. The results are from the following sources: regular languages [28] (for union of regular languages see also [21]), ideals [6], closed languages [7], prefix-free languages [14], suffix-free languages [13], and the other free languages [9]. Since the complexity of union for all four types of closed languages and for regular languages is mn, we have the same complexity for all four types of convex languages. Similarly, since the complexity of intersection for all four types of ideal languages and for regular languages is mn, we have the same complexity for all four types.

Table 3. Complexities of union and intersection

	$K \cup L$	$K \cap L$
right, two-sided, all-sided ideals	$mn-(m+n-2)$	mn
left ideals	mn	mn
prefix-, factor-, subword-closed	mn	$mn-(m+n-2)$
suffix-closed	mn	mn
prefix-free	$mn-2$	$mn-2(m+n-3)$
suffix-free	$mn-(m+n-2)$	$mn-2(m+n-3)$
bifix-, factor-, subword-free	$mn-(m+n)$	$mn-3(m+n-4)$
convex	mn	mn
regular	mn	mn

Table 4 shows the results for difference and symmetric difference.

Table 4. Complexities of difference and symmetric difference

	$K \setminus L$	$K \oplus L$
right, two-sided, all-sided ideals	$mn-(m-1)$	mn
left ideals	mn	mn
prefix-, factor-, subword-closed	$mn-(n-1)$	mn
suffix-closed	mn	mn
prefix-free	$mn-(m+2n-4)$	$mn-2$
suffix-free	$mn-(m+2n-4)$	$mn-(m+n-2)$
bifix-, factor-, subword-free	$mn-(2m+3n-9)$	$mn-(m+n)$
convex	mn	mn
regular	mn	mn

The sources for the difference operations are: regular languages [5], ideals [6], closed languages [7], and free languages [9]. Since the complexity of symmetric

difference for all four types of closed languages and for regular languages is mn, we have the same complexity for all four types.

Since the complexity of difference for suffix-closed languages and for regular languages is mn, we have the same complexity for suffix-convex languages. This leaves the difference of the other three types of convex languages, which we now address.

Proposition 3. *If K and L are prefix-convex (respectively, factor-convex or subword-convex) with $\kappa(K) = m$ and $\kappa(L) = n$, then $\kappa(K \setminus L) \leqslant mn$, and this bound is tight if $|\Sigma| \geqslant 2$.*

Proof. Let $\Sigma = \{a, b\}$, $K = (b^*a)^{m-1}\Sigma^* = \Sigma^* \sqcup a^{m-1}$ and $\overline{L} = (a^*b)^{n-1}\Sigma^* = \Sigma^* \sqcup b^{n-1}$. Then K and $\overline{L}$ are all-sided ideals and L is subword-closed. Thus both K and L are subword-convex, and we know from [6] that $\kappa(K \cap \overline{L}) = mn$; thus $\kappa(K \setminus L) = mn$. □

In summary, the complexities of all four boolean operations in all four classes of convex languages are mn.

7 Complexities of Product, Star and Reversal

Table 5 shows the results for product, star, and reversal. In the product column, k denotes the number of accepting quotients of K. The sources are: regular languages [21, 28], ideals [6], closed languages [7], prefix-free languages [14], suffix-free languages [13], and the other free languages [9]. The bounds for convex languages are still open.

Table 5. Complexities of product, star and reversal

	KL	L^*	L^R
right ideals	$m + 2^{n-2}$	$n + 1$	2^{n-1}
left ideals	$m + n - 1$	$n + 1$	$2^{n-1} + 1$
two-sided, all-sided ideals	$m + n - 1$	$n + 1$	$2^{n-2} + 1$
prefix-closed	$(m + 1)2^{n-2}$	$2^{n-2} + 1$	2^{n-1}
suffix-closed	$(m - k)n + k$	n	$2^{n-1} + 1$
factor-, subword-closed	$m + n - 1$	2	$2^{n-2} + 1$
prefix-free	$m + n - 2$	n	$2^{n-2} + 1$
suffix-free	$(m - 1)2^{n-2} + 1$	$2^{n-2} + 1$	$2^{n-2} + 1$
bifix-, factor-, subword-free	$m + n - 2$	$n - 1$	$2^{n-3} + 2$
regular	$m2^n - k2^{n-1}$	$2^{n-1} + 2^{n-2}$	2^n

8 Unary Convex Languages

Because product is commutative for unary languages, they have very special properties. Here, the concepts of prefix, suffix, factor, and subword all coincide. Therefore we can discuss convex unary languages in the terminology of prefix-convex languages. Let $\Sigma = \{a\}$, and suppose that $L \subseteq \Sigma^*$ is prefix-convex. If L is infinite and its shortest word is a^i, then $L = a^i a^*$. If L is finite, then $L = a^i \cup a^{i+1} \cup \cdots \cup a^j$, for $0 \leqslant i \leqslant j$.

If L is a unary right ideal with shortest word a^i, $i \geqslant 0$, then $L = a^i a^*$.

If L is unary and prefix-closed, then it is $L = a^*$, or $L = \{\varepsilon, a, \ldots, a^i\}$, $i \geqslant 0$.

If L is unary and prefix-free, then it is $L = a^i$, for some $i \geqslant 0$.

Table 6 shows the complexities of boolean operations. Here $\kappa(K) = m$, and $\kappa(L) = n$. The results for the union and intersection of regular unary languages are from [27]; for difference and symmetric difference see [5]. The bounds for boolean operations on convex languages are all $\max(m, n)$. For example, if $K = a^i \cup a^{i+1} \cup \cdots \cup a^{m-2}$ and $L = a^j \cup a^{j+1} \cup \cdots \cup a^{n-2}$, then $\kappa(K) = m$, $\kappa(L) = n$, and $\kappa(K \cup L) = \max(m, n)$. If $K = a^{m-1}a^*$ and $L = a^{n-1}a^*$, then $\kappa(K) = m$, $\kappa(L) = n$, and $\kappa(K \cap L) = \max(m, n)$.

Table 6. Complexity of boolean operations on unary convex languages

	$K \cup L$	$K \cap L$	$K \setminus L$	$K \oplus L$
unary ideal	$\min(m, n)$	$\max(m, n)$	n	$\max(m, n)$
unary closed	$\max(m, n)$	$\min(m, n)$	m	$\max(m, n)$
unary free	$\max(m, n)$	$m = n$	m	$\max(m, n)$
unary convex	$\max(m, n)$	$\max(m, n)$	$\max(m, n)$	$\max(m, n)$
unary regular	mn	mn	mn	mn

The complexities of product, star and reversal are given in Table 7. The results for these operations on regular unary languages are from [28]. The bound for the star of a unary convex language is derived below.

Table 7. Complexity of product, star and reversal on unary convex languages

	KL	L^*	L^R
unary ideal	$m + n - 1$	$n - 1$	n
unary closed	$m + n - 2$	2	n
unary free	$m + n - 2$	$n - 2$	n
unary convex	$m + n - 1$	$n^2 - 7n + 13$	n
unary regular	mn	$n^2 - 2n + 2$	n

Proposition 4. *If L is a unary convex language with $\kappa(L) = n$, then $\kappa(L^*) \leqslant n^2 - 7n + 13$, and the bound is tight if $n \geqslant 5$.*

Proof. If $L = \emptyset$, then $\kappa(L^*) = 2$.

If L is infinite, then it has the form $L = a^{n-1}a^*$, and $\kappa(L) = n$. If $n = 1$, then $L = a^* = L^*$, and $\kappa(L^*) = 1$. If $n = 2$, then $L = aa^*$ and $L^* = a^*$ again. For $n > 2$, $L^* = \varepsilon \cup a^{n-1}a^*$, and $\kappa(L^*) = n$.

Now consider the case where L consists of only one word. If $L = \varepsilon$, then $\kappa(L) = 2$, $L^* = \varepsilon$, and $\kappa(L^*) = 2$. If $L = a$, then $\kappa(L) = 3$, $L^* = a^*$, and $\kappa(L^*) = 1$. If $L = a^{n-2}$, for $n \geqslant 4$, then $\kappa(L) = n$, $L^* = (a^{n-2})^*$, and $\kappa(L^*) = n - 2$.

Next suppose that L is finite, and contains at least two words. Then L has the form $L = a^i \cup a^{i+1} \cup \cdots \cup a^{n-2}$. If $a \in L$, then $L^* = a^*$, and $\kappa(L^*) = 1$. Hence assume that $i \geqslant 2$, which implies that $n \geqslant 5$. The integers i and $j = i + 1$ are relatively prime, and the largest integer k that cannot be expressed as a non-negative linear combination of i and j is the Frobenius number [23] of i and j, which is $ij - i - j = i^2 - i - 1$. This means that $a^{i^2-i-1} \notin (a^i \cup a^j)^* \subseteq L^*$, and $a^{i^2-i}a^* \subseteq L^*$. If we add words of length greater than j, the length of the longest word not in L^* can only decrease. Hence the worst case for the complexity of L^* occurs when $L = a^{n-3} \cup a^{n-2}$. Here $\kappa(L^*) = (n-3)(n-2) - (n-3+n-2) + 2 = n^2 - 7n + 13$. □

9 Closed and Ideal Language Classes

Let $\mathbb{L}$ be the class of left ideals, and let $\mathbb{L}_\emptyset = \mathbb{L} \cup \{\emptyset\}$. The class $\mathbb{L}_\emptyset$ has been studied by Paz and Peleg [22] under the name *ultimate definite events*. They observed that, if I is some index set and L_i are left ideals, then $\bigcup_{i \in I} L_i$ and $\bigcap_{i \in I} L_i$ left ideals as well. They also showed the following:

Proposition 5. *The algebra $\mathcal{L} = (\mathbb{L}_\emptyset, \cup, \cap, \emptyset, \Sigma^*)$ is a complete lattice with least element $\emptyset$ and greatest element Σ^*. Moreover, $(\mathbb{L}_\emptyset, \cdot, \emptyset)$ is a semigroup with zero $\emptyset$, and $\mathbb{L}_\emptyset$ is closed under positive closure.*

Now let $\mathbb{S}$ be the class of suffix-closed languages. Then we have:

Proposition 6. *The algebra $\mathcal{L}' = (\mathbb{S}, \cap, \cup, \Sigma^*, \emptyset)$ is a complete lattice, and it is isomorphic to $\mathcal{L}$, with complementation acting as the isomorphism. Moreover, the algebra $(\mathbb{S}, \cdot, \{\varepsilon\}, \emptyset)$ is a monoid with unit $\{\varepsilon\}$ and zero $\emptyset$, and $\mathbb{S}$ is closed under star.*

Proposition 7. *The algebra $\mathcal{L}'' = (\mathbb{L}_\emptyset \cup \mathbb{S}, \cdot, \{\varepsilon\}, \emptyset)$ is a monoid with unit $\{\varepsilon\}$ and zero $\emptyset$, and $\mathbb{L}_\emptyset \cup \mathbb{S}$ is closed under complementation.*

Proof. We need only verify that $\mathcal{L}''$ is closed under product. First, suppose that one of K and L is $\emptyset$; then $KL = \emptyset$ and $\emptyset$ is in $\mathbb{L}_\emptyset \cap \mathbb{S}$. Hence assume that K and L are non-empty. Because we know that $\mathbb{L}$ and $\mathbb{S}$ are closed under product, we only need to consider the cases KL, where K is a left ideal and L is suffix-closed

or *vice versa*. In the first case, $KL = \Sigma^* KL$ is a left ideal. In the second case, since K is closed, it contains ε; thus $KL \supseteq L$. But we also have $KL \subseteq \Sigma^* L = L$. Hence $KL = L = \Sigma^* L$ is a left ideal. □

If $\mathbb{R}$ is the class of right ideals, let $\mathbb{R}_\emptyset = \mathbb{R} \cup \{\emptyset\}$, and let $\mathbb{P}$ be the class of prefix-closed languages. Then results analogous to Propositions 5–7 also hold for the algebras $(\mathbb{R}_\emptyset, \cup, \cap, \emptyset, \Sigma^*)$, $(\mathbb{R}_\emptyset, \cdot, \emptyset)$, $(\mathbb{P}, \cap, \cup, \Sigma^*, \emptyset)$, $(\mathbb{P}, \cdot, \{\varepsilon\}, \emptyset)$, and $(\mathbb{R}_\emptyset \cup \mathbb{P}, \cdot, \{\varepsilon\}, \emptyset)$. Similar statements hold for the class $\mathbb{T}$ of two-sided ideals with the class $\mathbb{F}$ of factor-closed languages, and the class $\mathbb{A}$ of all-sided ideals with the class $\mathbb{W}$ of subword-closed languages.

The close relationship between ideals and closed languages is reflected in quotient complexity. If we know the complexity $\kappa(K \cap L)$, where K and L are ideals, then we also know $\kappa(K \cap L) = \kappa(\overline{K} \cup \overline{L}) = \kappa(\overline{K} \cup \overline{L})$, where $\overline{K}$ and $\overline{L}$ are closed languages, and vice versa. Similar statements hold for the other boolean operations. Also, since reversal commutes with complementation, the complexities of the reversal of ideal languages are identical to those of closed languages. In Fig. 1, $\mathbb{PC}$, $\mathbb{SC}$, $\mathbb{BC}$, $\mathbb{FC}$, and $\mathbb{WC}$ stand for prefix-, suffix-, bifix-, factor-, and subword-convex languages, respectively.

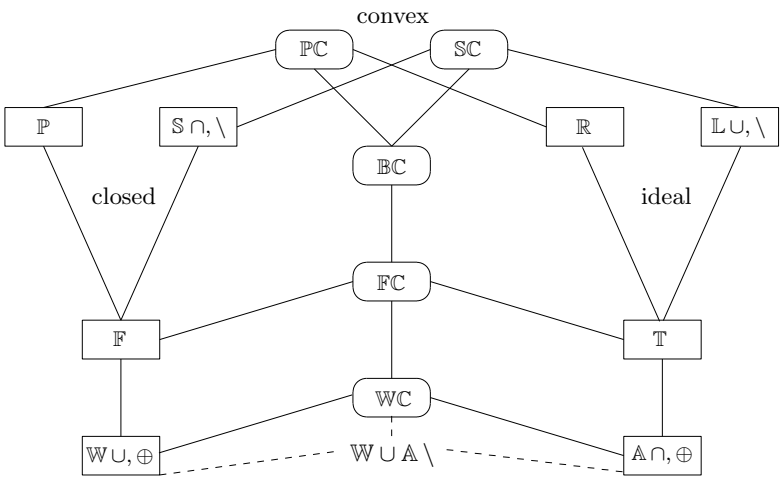

Fig. 1. Maximal complexities for boolean operations on convex languages.

The figure shows how the maximal bound mn for boolean operations can be reached in the convex hierarchy. For symmetric difference, it is reached at the bottom of both the closed and the ideal hierarchies. For union, it is reached in subword-closed languages in the closed hierarchy, but we have to go as high as left ideals in the ideal hierarchy. Intersection is dual to union, as shown in the figure. For difference, we can either go as high as suffix-closed languages or left ideals, or we can reach the bound in $\mathbb{W} \cup \mathbb{A}$, as witnessed by the languages in the proof of Proposition 3.

Since star and reversal are unary, the complexity of each of these operations in the union of a closed class with the corresponding ideal class is the maximum of the complexities in the closed and the ideal classes. This also holds for the product, as we now show.

Proposition 8. *If $K \subseteq \Sigma^*$ and $L \subseteq \Sigma^*$ are languages with $\kappa(K) = m$, $\kappa(L) = n$, and k is the number of accepting quotients of K; then the following hold:*

1. *If $K, L \in \mathbb{L}_\emptyset \cup \mathbb{S}$, then $\kappa(KL) \leqslant (m-k)n + k$, and the bound is tight in $\mathbb{S}$.*
2. *If $K, L \in \mathbb{R}_\emptyset \cup \mathbb{P}$, then $\kappa(KL) \leqslant (m+1)2^{n-2}$, and the bound is tight in $\mathbb{P}$.*
3. *If $K, L \in \mathbb{T}_\emptyset \cup \mathbb{F}$, then $\kappa(KL) \leqslant m+n-1$, and the bound is tight in $\mathbb{T}$ and $\mathbb{F}$.*
4. *If $K, L \in \mathbb{A}_\emptyset \cup \mathbb{W}$, then $\kappa(KL) \leqslant m+n-1$; the bound is tight in $\mathbb{A}$ and $\mathbb{W}$.*

Proof. The case when one of K and L is empty is trivial.

1. Consider the product of a left ideal K with a suffix-closed language L. It was shown in [7] that, for any language K and a suffix-closed language L, $\kappa(KL) \leqslant (m-k)n + k$, and that this bound is met by K and L that are both suffix-closed.
   In case K is suffix-closed and L is a left ideal, then $KL = L$, as we have shown in the proof of Proposition 7. Hence $\kappa(KL) = n$, which is less than the complexities in $\mathbb{L}$ and $\mathbb{S}$.

2. If K is a right ideal and L is prefix-closed, then $KL = K$ by an argument similar to that in the proof of Proposition 7. So $\kappa(KL) = m$, which is smaller than the bounds in $\mathbb{R}$ and $\mathbb{P}$.
   If K is prefix-closed and L is a right ideal, then $\varepsilon \in K$. Note that a prefix-closed language is either Σ^* or has $\emptyset$ as a quotient. In the first case, $KL = \Sigma^* L \Sigma^*$ and $\kappa(KL) \leqslant 2^{n-2} + 1$ [6], which is smaller than the bounds in $\mathbb{R}$ and $\mathbb{P}$. Now assume that $K \neq \Sigma^*$, which implies that K has $m-1$ accepting quotients and $\emptyset$.

   If K_w is accepting, then so is K_u for every prefix u of w. From Equation (2), $(KL)_w = K_w L \cup L_w \cup (\bigcup_{w=uv} L_v)$. Then each quotient of KL is determined by a quotient of K and a union of quotients of L that always contains L. Because L has Σ^* as a quotient, 2^{n-1} unions are equal to Σ^*. Thus for each accepting quotient of K we have 2^{n-2} possible unions plus Σ^*. Altogether there are at most $(m-1)2^{n-2} + 1$ quotients of KL of this type.

   If $K_w = \emptyset$ is the one rejecting quotient of K, then $(KL)_w$ is a union of quotients of L, which is non-empty because of L_w. Since unions that contain Σ^* have already been taken into account, we have $(2^{n-1} - 1)$ additional quotients, for a total of at most $(m+1)2^{n-2}$. This bound is reached in $\mathbb{P}$ [7].
3. If K is a two-sided ideal and L is factor-closed, then $KL = K$. If K is factor-closed and L is a two-sided ideal, then $KL = L$. In either case, this bound is lower than the bound in $\mathbb{T}$ and $\mathbb{F}$.

4. The argument of the last case works here as well. □

Given any language L of complexity n, it is natural to ask what is the worst-case complexity of the prefix-closure of L and the converse prefix-closure of L, which is the right ideal generated by L. The same questions apply to the other closed languages. The results [6, 18] for these two closures are summarized in Table 8.

The complexities of ideals in terms of their minimal generators and of minimal generators in terms of ideals are also discussed in [6].

Table 8. Complexities of closure and converse closure

	closure	converse closure
prefix relation	n	n
suffix relation	$2^n - 1$	2^{n-1}
factor relation	2^{n-1}	$2^{n-2} + 1$
subword relation	$2^{n-2} + 1$	$2^{n-2} + 1$

10 Conclusions

It has been demonstrated that the class of convex languages and its subclasses are interesting from several points of view, and that studying these classes together is very worthwhile.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant No. OGP000871.

I am grateful to Galina Jirásková for a very careful reading of the manuscript, and to Jeff Shallit for the reference to Frobenius numbers.

References

1. Ang, T., Brzozowski, J.: Languages convex with respect to binary relations, and their closure properties. Acta Cybernet. **19**(2) (2009) 445–464
2. Berstel, J., Perrin, D.: Theory of Codes. Academic Press (1985)
3. Berstel, J., Perrin, D., Reutenauer, C.: Codes and Automata (Encyclopedia of Mathematics and its Applications). Cambridge University Press (2010)
4. Brzozowski, J.: Derivatives of regular expressions. J. ACM **11**(4) (1964) 481–494
5. Brzozowski, J.: Quotient complexity of regular languages. In Dassow, J., Pighizzini, G., Truthe, B., eds.: Proceedings of the 11th International Workshop on Descriptional Complexity of Formal Systems, Magdeburg, Germany, Otto-von-Guericke-Universität (2009) 25–42 (Extended abstract at `http://arxiv.org/abs/0907.4547`).

6. Brzozowski, J., Jirásková, G., Li, B.: Quotient complexity of ideal languages. In López-Ortiz, A., ed.: Proceedings of the 9th Latin American Theoretical Informatics Symposium, (LATIN). Volume 6034 of LNCS, Springer (2010) 208–211 (Full paper at `http://arxiv.org/abs/0908.2083`).
7. Brzozowski, J., Jirásková, G., Zou, C.: Quotient complexity of closed languages. In: Proceedings of the 5th International Computer Science Symposium in Russia, (CSR). LNCS, Springer (2010) To appear. (Preprint at `http://arxiv.org/abs/0912.1034`).
8. Brzozowski, J., Shallit, J., Xu, Z.: Decision problems for convex languages. In Dediu, A.H., Ionescu, A.M., Martín-Vide, C., eds.: LATA 2009. Volume 5457 of LNCS, Springer (2009) 247–258
9. Brzozowski, J., Smith, J.: Quotient complexity of bifix-, factor-, and subword-free languages. In preparation.
10. Câmpeanu, C., Culik II, K., Salomaa, K., Yu, S.: State complexity of basic operations on finite languages. In Boldt, O., Jürgensen, H., eds.: Revised Papers from the 4th International Workshop on Automata Implementation, (WIA). Volume 2214 of LNCS, Springer (2001) 60–70
11. Gill, A., Kou, L.T.: Multiple-entry finite automata. J. Comput. Syst. Sci. **9**(1) (1974) 1–19
12. Haines, L.: On free monoids partially ordered by embedding. J. Combin. Theory **6**(1) (1969) 94–98
13. Han, Y.S., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theoret. Comput. Sci. **410**(27-29) (2009) 2537–2548
14. Han, Y.S., Salomaa, K., Wood, D.: Operational state complexity of prefix-free regular languages. In Ésik, Z., Fülöp, Z., eds.: Automata, Formal Languages, and Related Topics, University of Szeged, Hungary (2009) 99–115
15. Higman, G.: Ordering by divisibility in abstract algebras. Proc. London Math. Soc. **3**(2) (1952) 326–336
16. Jürgensen, H., Konstantinidis, S.: Codes. In Rozenberg, G., Salomaa, A., eds.: Handbook of Formal Languages, Volume 1: Word, Language, Grammar. Springer (1997) 511–607
17. Jürgensen, H., Yu, S.S.: Relations on free monoids, their independent sets, and codes. Internat. J. Comput. Math. **40** (1991) 17–46
18. Kao, J.Y., Rampersad, N., Shallit, J.: On NFAs where all states are final, initial, or both. Theoret. Comput. Sci. **410**(47-49) (2009) 5010–5021
19. Kruskal, J.B.: The theory of well-quasi-ordering: A frequently discovered concept. J. Combin. Theory A **13**(3) (1972) 297–305
20. de Luca, A., Varricchio, S.: Some combinatorial properties of factorial languages. In Capocelli, R., ed.: Sequences: Combinatorics, Compression, Security, and Transmission. Springer (1990) 258–266
21. Maslov, A.N.: Estimates of the number of states of finite automata. Dokl. Akad. Nauk SSSR **194** (1970) 1266–1268 (Russian) English translation: Soviet Math. Dokl. **11** (1970), 1373–1375.
22. Paz, A., Peleg, B.: Ultimate-definite and symmetric-definite events and automata. J. ACM **12**(3) (1965) 399–410
23. Ramírez Alfonsín, J.L.: The Diophantine Frobenius Problem. Volume 30 of Oxford Lectures Series in Mathematics and its Applications. Oxford University Press (2005)
24. Shyr, H.J.: Free Monoids and Languages. Hon Min Book Co, Taiwan (2001)
25. Shyr, H.J., Thierrin, G.: Hypercodes. Information and Control **24** (1974) 45–54

26. Thierrin, G.: Convex languages. In Nivat, M., ed.: Automata, Languages and Programming. North-Holland (1973) 481–492
27. Yu, S.: State complexity of regular languages. J. Autom. Lang. Comb. **6** (2001) 221–234
28. Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theoret. Comput. Sci. **125**(2) (1994) 315–328